

# Image Classification Game

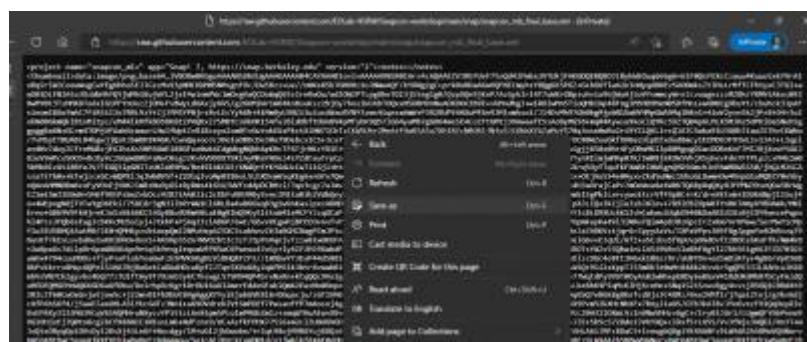
This **Snap! game** uses **Nvidia Jetson** capability to classify images.

## Offline Snap! downloading

Please download and open Offline version of Snap! for our project. Go to <https://snap.berkeley.edu/offline> and follow the steps.

## Snap! files' downloading

Please open the link [Classification Game](#) to download our project on your computer. Probably you would see the xml in raw format. Click the right button of your mouse and save it on the disk.



## Web camera Image in Snap!

You can get picture from your web camera in Snap!.

- **video capture** block to enable video capturing.



- Change value of **set video transparency** block to 0 for clear image.



- **video snap on stage** block reports picture from stage.



## Connection to Jetson from Snap!

If you have not imported it yet, please download [jetson blocks](#) and import it to your Snap! project.





following blocks.

- Use **connect to Jetson url** block to connect Jetson.

 ws://ip\_address:4040



You need **ip address** of Nvidia Jetson. You can use *ifconfig* command in a terminal to get **ip address**.

- Store the value of **jetson\_name** in a variable.

 set jetson\_name to dev-kit-20

- Store the value of **connect to Jetson url** block in a variable for later use.

 set jetson to connect to jetson url join ws://jetson\_name:4040

## Response from classification

Here we will send **video snap on stage** to Jetson for processing. Jetson will respond back class name, confidence value and class ID.



Only **class name** and **confidence value** will be used in this example. This project does not use **class ID**.

- Use **get response from Jetson** block to send **image**, and get **class name** and **confidence value**.

- First input slot is for **jetson** variable that stores websocket data.
- Second input slot is for **costume** you want to be classified by Nvidia Jetson.

 set response to send picture to jetson jetson video snap on Stage

## Class name and confidence value

This section will demonstrate how to handle **response** variable to access **class name** and **confidence value**.

- **class name** is the 2nd item of 1st item of **response** block.

 item 2 of item 1 of response wig

- **confidence value** is 2nd item of 2nd item of **response** block.



```
round (100 × item (2 of item (2 of response))
```

Multiply **confidence** value by 100 to get percentage of **confidence** .

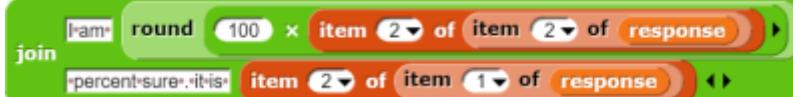


You can create custom blocks, to get **class name** *get class name from response* and to get **confidence** *get confidence from response* .

## Speech functionality

**Speech functionality** is available as a library in Snap!. Select *export libraries* from settings then choose *speech module* .

- Use **join** block to create text like *I am confidence percent sure, it is class name* .



```
join (I-am round (100 × item (2 of item (2 of response))) + (percent-sure-it-is item (2 of item (1 of response)))
```

- Use **speak** and **wait** block to read text a loud.

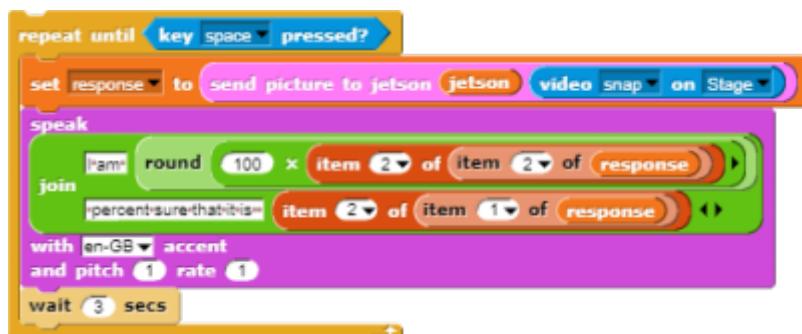


```
speak (join (I-am round (100 × item (2 of item (2 of response))) + (percent-sure-it-is item (2 of item (1 of response))) with en-GB accent and pitch 1 rate 1) and wait)
```

## Repeat block for game

Last step is adding loop for the game.

- Use repeat block and put script inside of it.



```
repeat until (key space pressed? [set response to (send picture to jetson jetson video snap on Stage) speak (join (I-am round (100 × item (2 of item (2 of response))) + (percent-sure-that-it-is item (2 of item (1 of response))) with en-GB accent and pitch 1 rate 1) and wait (3 secs))])
```



This example used **repeat until** block to break loop when space key pressed.

You can download full game from [Github page of EOLab-HSRW](#).

From:

<https://student-wiki.eolab.de/> - **HSRW EOLab Students Wiki**

Permanent link:

<https://student-wiki.eolab.de/doku.php?id=snapcon2022:image-classification-game&rev=1659786975>

Last update: **2023/01/05 14:38**

